

1 **CLAIMS**

2 **1.** A method comprising:

3 sorting, using multiple depth buffers, depth data associated with multiple
4 transparent pixels that overlie one another to identify an individual pixel that lies
5 closest to an associated opaque pixel;

6 computing a transparency effect of the identified pixel relative to the
7 associated opaque pixel; and

8 after said computing, identifying a next closest transparent pixel relative to
9 the opaque pixel and computing, for the next closest pixel, a transparency effect
10 relative to the transparency effect that was just computed.

11
12 **2.** The method of claim 1, wherein said multiple depth buffers comprise z
13 buffers.

14
15 **3.** The method of claim 1, wherein said act of sorting comprises:

16 identifying one of the multiple buffers as a destination buffer that is both
17 readable and writable;

18 identifying another of the multiple buffers as a source buffer that is only
19 readable; and

20 flipping which of the multiple buffers is considered as the destination
21 buffer and the source buffer during said acts of sorting, computing and identifying.

22
23 **4.** The method of claim 1 further comprising repeating said act of identifying
24 for any additional overlying transparent pixels.

1 5. A computing system configured to implement the method of claim 1.

2
3 6. An apparatus comprising:

4 means for sorting, using multiple depth buffers, depth data associated with
5 multiple transparent pixels that overlie one another to identify an individual pixel
6 that lies closest to an associated opaque pixel;

7 means for computing a transparency effect of the identified pixel relative to
8 the associated opaque pixel; and

9 means for identifying a next closest transparent pixel relative to the opaque
10 pixel and computing, for the next closest pixel, a transparency effect relative to the
11 transparency effect that was just computed.

12
13 7. The apparatus of claim 6, wherein said means for sorting and means for
14 identifying comprises hardware comparison logic.

1 **8.** A method comprising:

2 (a) rendering at least one opaque pixel that lies along a ray;

3 (b) identifying a transparent pixel that lies along the ray, the identified
4 transparent pixel being the closest transparent pixel to the opaque pixel;

5 (c) computing transparency effects of the identified transparent pixel
6 relative to the opaque pixel;

7 (d) if additional transparent pixels lie along the ray, identifying a next
8 closest transparent pixel relative to the opaque pixel and computing transparency
9 effects of the next closest pixel relative to the computed transparency effects of a
10 last computed transparent pixel; and

11 (e) repeating act (d) until transparency effects of all of the transparent
12 pixels along the ray have been computed in a back-to-front manner.

13
14 **9.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
15 physical depth buffers for sorting depth data associated with the transparent pixels.

16
17 **10.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
18 physical depth buffers for sorting depth data associated with the transparent pixels,
19 and wherein the two depth buffers are configured to be flipped.

20
21 **11.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
22 physical z buffers for sorting depth data associated with the transparent pixels.

1 **12.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
2 physical z buffers for sorting depth data associated with the transparent pixels, and
3 wherein the two z buffers are configured to be flipped.

4
5 **13.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
6 physical depth buffers for sorting depth data associated with the transparent pixels,
7 and wherein performing acts (a)-(e) comprise:

8 designating one of the depth buffers as readable and writable;
9 designating the other of the depth buffers as readable only; and
10 flipping the designations of the depth buffers.

11
12 **14.** The method of claim 8, wherein acts (a)-(e) are performed utilizing two
13 physical z buffers for sorting depth data associated with the transparent pixels, and
14 wherein performing acts (a)-(e) comprise:

15 designating one of the z buffers as readable and writable;
16 designating the other of the z buffers as readable only; and
17 flipping the designations of the z buffers.

18
19 **15.** A computing system configured to implement the method of claim 8.
20
21
22
23
24
25

1 **16.** A system comprising:

2 means for rendering at least one opaque pixel that lies along a ray;

3 means for identifying a transparent pixel that lies along the ray, the
4 identified transparent pixel being the closest transparent pixel to the opaque pixel;

5 means for computing transparency effects of the identified transparent pixel
6 relative to the opaque pixel; and

7 means for identifying, in a back-to-front manner, additional transparent
8 pixels and successively computing transparency effects for the additional
9 transparent pixels.

10
11 **17.** The system of claim 16, wherein said means for rendering comprises a
12 graphics subsystem.

13
14 **18.** The system of claim 16, wherein said means for identifying comprises a
15 pair of physical depth buffers.

16
17 **19.** The system of claim 16, wherein said means for identifying comprises a
18 pair of physical depth buffers that can be logically flipped.

1 **20.** A system comprising:
2 a transparent depth sorting component comprising:
3 at least two physical depth buffers;
4 a writeback counter to count writebacks that occur to at least one of
5 the two physical depth buffers; and
6 comparison logic that is configured to effect:
7 sorting, using said at least two physical buffers, of depth data
8 associated with multiple transparent pixels that overlie one another
9 to identify an individual pixel that lies closest to an associated
10 opaque pixel;
11 computing a transparency effect of the identified pixel
12 relative to the associated opaque pixel;
13 after said computing, identifying a next closest transparent
14 pixel relative to the opaque pixel; and
15 computing, for the next closest pixel, a transparency effect
16 relative to the transparency effect that was computed for the said
17 closest individual pixel and the associated opaque pixel.

1 **21.** The system of claim 20, wherein:

2 one of said at least two physical depth buffers is capable of being
3 designated as readable and writable;

4 another of said at least two physical depth buffers is capable of being
5 designated as readable only; and

6 designations of said at least two physical depth buffers are capable of being
7 flipped.

8
9 **22.** The system of claim 20, wherein said at least two physical depth buffers
10 comprise z buffers.

11
12 **23.** The system of claim 20, wherein said transparent depth sorting component
13 is configured to terminate transparent depth sorting when the writeback counter
14 indicates that no writebacks have occurred.

15
16 **24.** A graphics subsystem embodying the transparent depth sorting component
17 of claim 20.

18
19 **25.** A computer system embodying the graphics subsystem of claim 24.
20
21
22
23
24
25

1 **26.** A method comprising:

2 mapping a first of two depth buffers as a destination buffer that is readable
3 and writable, a second of the two depth buffers being designated as a source buffer
4 that is only readable;

5 rendering one or more opaque objects having associated opaque pixels;

6 writing a depth value associated with an opaque pixel to the first buffer;

7 mapping the second of the depth buffers as the destination buffer, the first
8 of the depth buffers being designated as the source buffer;

9 initializing the destination buffer to a predetermined value;

10 effecting a comparison of a new pixel depth value with values in the source
11 and destination buffers and writing the new pixel depth value to the destination
12 buffer if the new pixel depth value is (a) greater than the value currently in the
13 destination buffer and (b) less than the value in the source buffer, effective to write
14 a new pixel depth value that is associated with a pixel that is closest to a pixel
15 whose depth value is contained in the source buffer;

16 rendering one or more transparent objects having associated transparent
17 pixels;

18 determining if transparency effects for all transparent pixels along an
19 associated ray have been accounted for and if so, terminating processing for pixels
20 that lie along the ray and, if not:

21 mapping the first of the depth buffers as the destination buffer, the
22 second of the buffers being designated as the source buffer;

23 effecting a comparison of the new pixel depth value with values in
24 the source and destination buffers and writing to a frame buffer and the
25 destination buffer if the new pixel depth value is equal to the value in the

1 source buffer and the new pixel depth value is less than the value in the
2 destination buffer;

3 rendering one or more transparent objects; and

4 returning to said act of mapping the second of the depth buffers until
5 transparency effects of all transparent pixels that lie along the ray have been
6 accounted for.

7
8 **27.** The method of claim 26, wherein said predetermined value comprises a
9 depth buffer's smallest value.

10
11 **28.** The method of claim 26, wherein said act of determining is performed by
12 maintaining a depth buffer writeback counter that keeps track of depth buffer
13 writebacks.

14
15 **29.** The method of claim 26, wherein the depth buffers comprise z buffers.

16
17 **30.** A computing system configured to implement the method of claim 26.

18
19 **31.** A system comprising:

20 a processor;

21 at least two depth buffers;

22 a frame buffer; and

23 a graphics subsystem operably connected with the processor and configured
24 to, under the influence of the processor:

1 map a first of the depth buffers as a destination buffer that is
2 readable and writable, a second of the depth buffers being designated as a
3 source buffer that is only readable;

4 render one or more opaque objects having associated opaque pixels;
5 write a depth value associated with an opaque pixel to the first
6 buffer;

7 map the second of the depth buffers as the destination buffer, the
8 first of the depth buffers being designated as the source buffer;

9 initialize the destination buffer to a predetermined value;

10 effect a comparison of a new pixel depth value with values in the
11 source and destination buffers and write the new pixel depth value to the
12 destination buffer if the new pixel depth value is (a) greater than the value
13 currently in the destination buffer and (b) less than the value in the source
14 buffer, effective to write a new pixel depth value that is associated with a
15 pixel that is closest to a pixel whose depth value is contained in the source
16 buffer;

17 render one or more transparent objects having associated transparent
18 pixels;

19 determine if transparency effects for all transparent pixels along an
20 associated ray have been accounted for and if so, terminate processing for
21 pixels that lie along the ray and, if not:

22 map the first of the depth buffers as the destination buffer, the
23 second of the buffers being designated as the source buffer;

24 effect a comparison of the new pixel depth value with values
25 in the source and destination buffers and write to the frame buffer

1 and the destination buffer if the new pixel depth value is equal to the
2 value in the source buffer and the new pixel depth value is less than
3 the value in the destination buffer;

4 render one or more transparent objects; and

5 return to said mapping the second of the depth buffers until
6 transparency effects of all transparent pixels that lie along the ray
7 have been accounted for.

8
9 **32.** The system of claim 31, wherein said predetermined value comprises a
10 depth buffer's smallest value.

11
12 **33.** The system of claim 31 further comprising a depth buffer writeback
13 counter that keeps track of depth buffer writebacks.

14
15 **34.** The system of claim 31, wherein the depth buffers comprise z buffers.
16
17
18
19
20
21
22
23
24
25